Using Reinforcement Learning to Estimate Human Joint Moments From Electromyography or Joint Kinematics: An Alternative Solution to Musculoskeletal-Based Biomechanics

Wen Wu
Joint Department of Biomedical Engineering,
University of North Carolina at Chapel Hill/North Carolina State University,
Raleigh, NC 27695
e-mail: melb.ww@gmail.com

Katherine R. Saul
Department of Mechanical and Aerospace Engineering,
North Carolina State University,
Raleigh, NC 27695
e-mail: ksaul@ncsu.edu

He (Helen) Huang[1]
Joint Department of Biomedical Engineering,
University of North Carolina at Chapel Hill/North Carolina State University,
Raleigh, NC 27695
e-mail: hhuang11@ncsu.edu

Reinforcement learning (RL) has potential to provide innovative solutions to existing challenges in estimating joint moments in motion analysis, such as kinematic or electromyography (EMG) noise and unknown model parameters. Here, we explore feasibility of RL to assist joint moment estimation for biomechanical applications. Forearm and hand kinematics and forearm EMGs from four muscles during free finger and wrist movement were collected from six healthy subjects. Using the proximal policy optimization approach, we trained two types of RL agents that estimated joint moment based on measured kinematics or measured EMGs, respectively. To quantify the performance of trained RL agents, the estimated joint moment was used to drive a forward dynamic model for estimating kinematics, which was then compared with measured kinematics using Pearson correlation coefficient. The results demonstrated that both trained RL agents are feasible to estimate joint moment for wrist and metacarpophalangeal (MCP) joint motion prediction. The correlation coefficients between predicted and measured kinematics, derived from the kinematics-driven agent and subject-specific EMG-driven agents, were 98% ± 1% and 94% ± 3% for the wrist, respectively, and were 95% ± 2% and 84% ± 6% for the metacarpophalangeal joint, respectively. In addition, a biomechanically reasonable joint moment-angle-EMG relationship (i.e., dependence of joint moment on joint angle and EMG) was predicted using only 15 s of collected data. In conclusion, this study illustrates that an RL approach can be an alternative technique to conventional inverse dynamic analysis in human biomechanics study and EMG-driven human-machine interfacing applications.
[DOI: 10.1115/1.4049333]

Keywords: reinforcement learning, musculoskeletal model, inverse dynamics, EMG-driven model

Introduction

Estimating joint moment is one of the most common biomechanical analyses, essential for computing muscle forces and internal joint contact forces [1], serving as a controlling input to human–machine interface (HMI) tools [2], and providing insight into the functional capacity of human joints [3]. Musculoskeletal (MSK) models that employ inverse dynamics and Hill-type musculotendon models have long been used to estimate joint moments from measured kinematics, kinetics, and electromyographic signals [4], and have contributed significantly to understanding of human biomechanics.

Challenges in application of these methods remain, especially regarding simulation accuracy, data availability, and signal quality. For example, inverse dynamics analysis can theoretically estimate joint moments of an MSK model from measured joint kinematics, either with or without external force measurements (e.g., ground reaction force). However, when external force measurements are unavailable, the accuracy of the estimation is greatly compromised because the second-order differentiation of the measured kinematics will amplify the measurement errors [5]. In addition, when applying estimated joint moment for forward dynamic simulation/control, these errors together with the computational errors in inverse and forward dynamics processes lead to significant drift of kinematics from measurements, which requires additional strategies to compensate and stabilize the simulation/control [6]. Moments can also be predicted using electromyography (EMG)-driven MSK models, with applications in HMI tools that drive virtual objects or robotic limbs [2,7]. However, this approach has two main challenges. First, the performance of EMG-driven MSK models relies on EMG signal quality. Unfortunately, surface EMG recordings are often contaminated with noise, such as motion artifacts and crosstalk [8]. In particular, EMG-driven hand movement can be more sensitive to EMG noise because of small muscle size, high motor unit density, and low-level activations during free hand movement [9]. To alleviate the effects of measurement noise, one solution is to employ optimization approaches to adjust EMG excitation signals to better predict joint moment [10]. Second, accurate modeling of musculotendon parameters is critical for reliable performance of an EMG-driven MSK model [11], but estimation of subject-specific musculotendon parameters (e.g., optimal muscle fiber length, tendon slack length, maximal muscle force) is difficult. Many studies estimate these parameters to minimize differences between estimated and measured isometric and isokinetic moments [12,13]; however, this method limits calibration tasks to constrained movement on a dynamometer and may not generalize to free movement. An alternative approach is to measure joint kinematics and synchronized EMG signals in free movements and use optimization to select subject-specific musculotendon parameters by minimizing the differences between measured and simulated kinematics [2], but this approach can be computationally expensive due to the optimization of forward-dynamics simulations required.

Reinforcement learning (RL) is an advanced machine learning method that has been used to tackle many challenging applications, such as control sophisticated robotics [14–16] and making programs that outperform top human players in decision-making games [17]. Compared to other data-driven approaches such as supervised learning and unsupervised learning that passively learn from the input data, the RL inherently reflects how humans and other animals learn in real world environments—it actively explores the given environment and learns to achieve long-term goals via rewarding desired actions or punishing undesired ones [18]. Additionally, the RL signifies the sequential effect in a series of decision-making—the decision at each time-step depends on previous decisions, while the outputs of supervised and unsupervised learnings are independent. Furthermore, RL is able to find solutions without requiring predefined knowledge if the agent can sufficiently explore the input domain of the environment [19]. Due to these advantages, there has been increasing number of RL

[1]Corresponding author.
Manuscript received February 23, 2020; final manuscript received November 27, 2020; published online February 1, 2021. Assoc. Editor: Sara Wilson.

applications in the field of human biomechanics for MSK model simulation, such as training a rigid-body MSK model to run and avoid obstacles [20–22], and assistive device control to change joint dynamics, such as learning optimal control of prosthetic legs [23,24] and control of a function electrical stimulation system for arm movement assistance [25]. However, to our knowledge, the number of studies that use RL to estimate joint moments via kinematics or EMG signals to assist human biomechanics study has been quite limited.

Hence, in this paper, we aimed to develop and evaluate a RL-based framework to (1) solve the problem of inverse dynamics for joint moment estimation, and (2) predict joint moments using EMG signals for future HMI applications. In addition, we examined whether the trained RL policies were able to reveal biomechanically reasonable joint moment-angle-EMG relationship (i.e., dependence of joint moment on joint angle and EMG) for specific human subjects. The results of this study led to novel alternative solutions to the conventional MSK-based approaches for joint moment estimation for various biomechanical applications.

Method

Data Collection and Kinetic Hand Model. Details regarding data collection and the kinetic hand model were presented in our previous publication [2]. Briefly, six healthy subjects were recruited with Institutional Review Board approval and instructed to flex or extend the wrist and metacarpophalangeal joints (MCP) of their dominant arm at self-selected directions and varied range of speeds, with the shoulder at zero abduction and elbow flexed at 90 deg. Each subject performed two trials for 30 s and rested between trials. Joint kinematics and surface EMGs of the *extensor digitorum*, *extensor carpi radialis longus, flexor digitorum,* and *flexor carpi radialis* were simultaneously collected at 120 Hz and 960 Hz, respectively. A kinetic hand model, including three rigid bodies (forearm, hand, and lumped-finger) and two hinge joints (i.e., wrist and MCP), was developed on the Unity 3D platform (Unity Technologies, San Francisco, CA) [26].

Agent–Environment Interaction. This study employed two RL agents—kinematics-driven and EMG-driven. Each was used to estimate the joint moment required to drive the kinetic hand model to replicate measured kinematics as closely as possible.

Kinematics-Driven Agent. The kinematics-driven agent determined joint moment from measured kinematics without using external force measurements. The agent policy was regulated by an artificial neural network with two hidden layers and 128 units in each hidden layer. The weight of each unit was randomly initialized from a truncated normal distribution centered on zero, and the bias was set to zero. The activation function employed the Swish function [27]. There were 16 inputs to the artificial neural network (details below), while the two outputs were joint moments of the wrist and MCP. Because a single kinetic hand model was used for all subjects, we trained one generic kinematics-driven agent using a single 30 s kinematics dataset that was collected from an arbitrarily chosen subject.

For a given time-step t, four simulated and 14 measured input states were passed to the agent (Fig. 1, agent environment integration block). The four simulated input states were joint angle (α_t) and joint angular velocity ($\dot{\alpha}_t$) of each joint of the kinetic hand model obtained from prior time-step. The 14 measured input states were measured angles of each joint at current (β_t) and future timesteps (β_{t+c1}, β_{t+c2}, …, β_{t+c6}), where $c1$, $c2$, …, and $c6$ were 0.2 s, 0.4 s, 0.6 s, 0.8 s, 1 s, and 1.2 s for each joint. Based on the RL policy and states, the RL agent determined optimal joint moment (T_t). This joint moment was then used with the kinetic hand model in a forward dynamics simulation to generate simulated states for the next time-step (i.e., α_{t+1} and $\dot{\alpha}_{t+1}$).

EMG-Driven Agent. The EMG-driven agent predicted joint moment from measured EMG. The setup was identical to the kinematics-driven agent, except (1) we trained subject-specific EMG-driven agents for each of the six subjects, because each subject inherently had different EMG magnitudes and crosstalk artifacts for a given joint angle and joint torque; (2) they were trained with a shorter data collection (15 s), thus reducing computational cost of forward dynamics in each iteration; (3) for each time-step, measured inputs to the agent only contained four EMG channels at the current time-step without any future insights because predictions were intended to mimic use in real-time HMI, making it eight inputs in total (i.e., four simulated inputs obtained from prior time-step + 4 measured inputs).

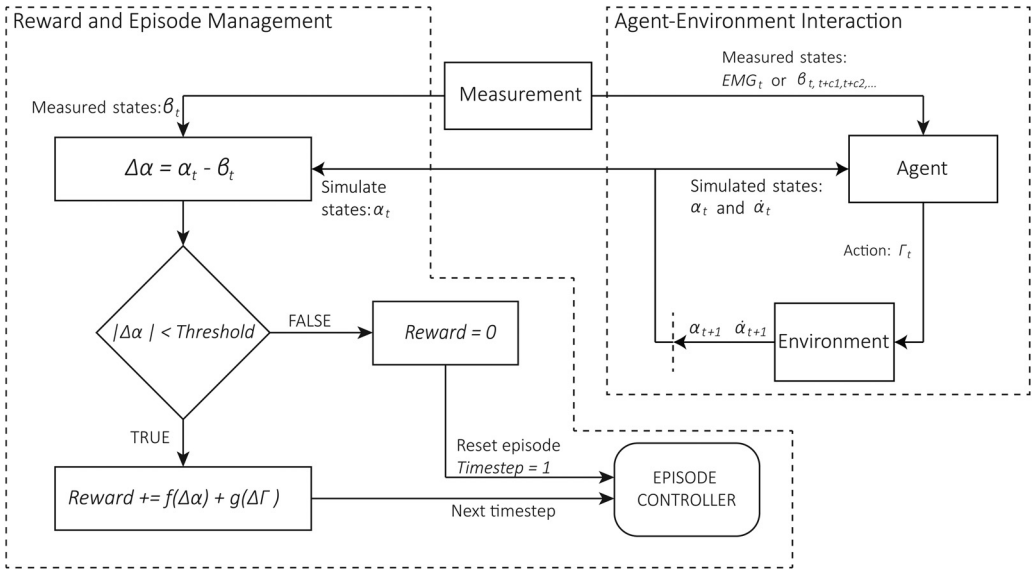

Fig. 1 RL flow diagram illustrating agent-environment integration and reward and episode management blocks. The agent-environment integration block includes an agent that acts according to the RL policy and a kinetic hand model that runs forward dynamics to predict the states of next time-step. The Reward and Episode Management block regulates the reward with respect to Eqs. (1) and (2), and controls the time-step of each episode.

Table 1 Hyperparameters used during training

Training hyperparameters	Value
Batch size	2048
Beta	1.50×10^{-2}
Buffer size	40960
Epsilon	0.2
Gamma	0.96
Lambda	0.95
Learning rate	1.00×10^{-4}
Normalize	True
Epoch number	3
Horizon time	64

The hyperparameters were defined in Schulman et al. [28] and Juliani et al. [26]. Specifically, "Batch size" is the number of experiences in each iteration of gradient descent; "Beta" is the strength of entropy regularization; "Buffer size" is the number of experiences to collect before updating the policy model; "Epsilon" influenced how rapidly the policy can evolve during training; "Gamma" is the reward discount rate; "Lambda" is the regularization parameter; "Learning rate" is the initial learning rate for gradient descent; "Normalize" indicates whether to automatically normalize observations; "Epoch number" is the number of passes to make through the experience buffer when performing gradient descent optimization; "Horizon time" indicates how many steps of experience to collect per agent before adding it to the experience buffer.

Reward and Episode Management. The initial reward of each episode (i.e., a sequence of states-action-reward from the initial state to the terminal state) was to set to 0. For a given time-step, t, absolute differences ($\Delta\alpha$) between simulated (α_t) and measured joint angles (β_t) were compared to an error threshold, 15 deg (Fig. 1, Reward and Episode Management block). If the absolute joint differences exceeded the threshold, the episode reset to time-step 1 and the reward reset to 0. If absolute joint differences were smaller than the threshold, the episode accumulated reward according to

$$\text{reward}_t = \text{reward}_{t-1} + f(\Delta\alpha) + g(\Delta\Gamma) \tag{1}$$

where $\Delta\Gamma$ was the difference between the moment at time-step $t-1$ and the moment at time-step t; both function $f(x)$ and $g(x)$ were of the form

$$f(x), g(x) = \frac{a}{|x| + b} \tag{2}$$

where a and b of $f(x)$ were both 0.3, whereas a and b of $g(x)$ were both 0.1 and 0.3, respectively. $f(x)$ granted greater rewards for smaller absolute error between simulated and measured joint angles, and $g(x)$ granted greater rewards for less fluctuated joint moments to simulate that human joint moments are typically continuous without sudden changes.

Reinforcement Learning Training. A free open-source RL toolbox—unity machine learning agents [26], employing the proximal policy optimization algorithm [28]–was used to update the optimal RL policy for each agent. All trainings were performed on a desktop computer with AMD Ryzen-7 1800X processor and 16-GB-RAM. Training hyperparameters are in Table 1.

The training procedures were ended by researchers when agents were able to finish the whole training dataset without reset and when there was no significant reward increase. During training, a single policy controlled 20 agents that ran forward dynamics in parallel. Though the 20 agents shared a single policy and same measured dataset, actions of each agent varied during a training session because of entropy regularization [28], thus increasing

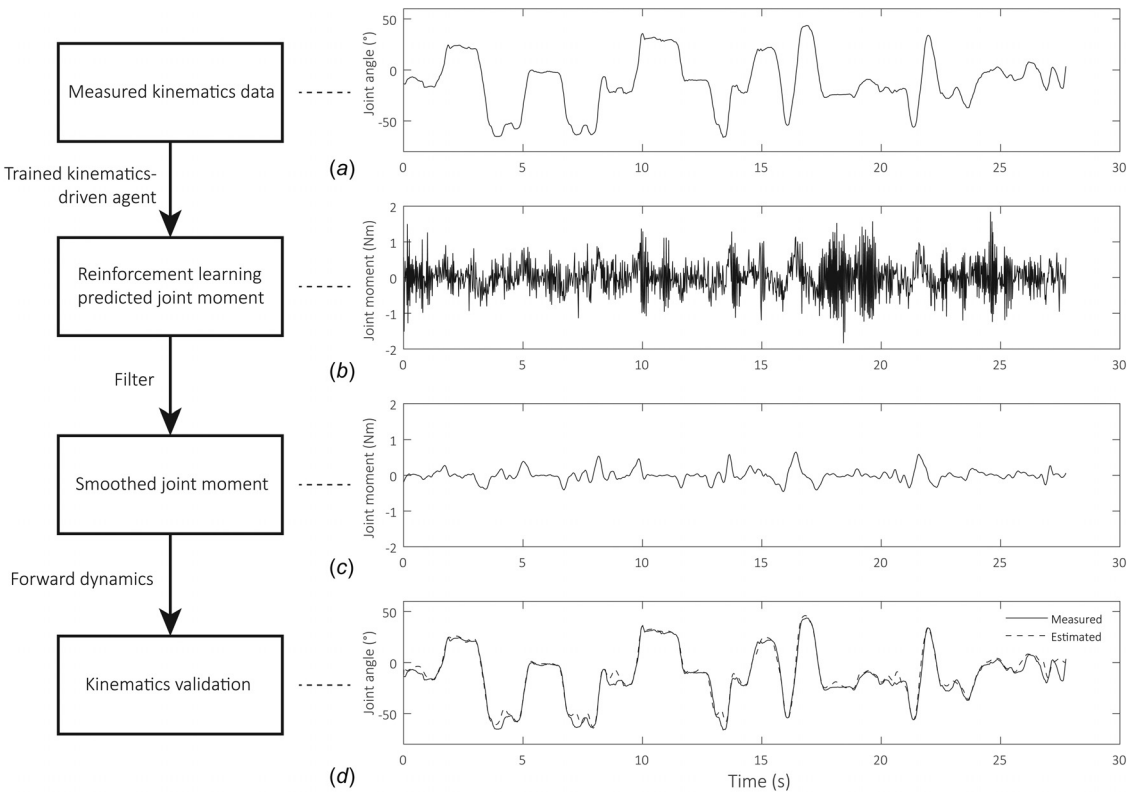

Fig. 2 The kinematics-driven agent cross-validation workflow using an example wrist joint dataset. Specifically, measured kinematics data (*a*) were passed to the trained agent to predict joint moments (*b*). The joint moments (*b*) were then smoothed (*c*) using a local regression filter. Finally, the smoothed joint moment (*c*) was validated by rerunning forward dynamics to predict joint kinematics and comparing to the original kinematics data (*d*).

Table 2 The RMSE error and the correlation coefficient of the wrist joint and MCP joint between the measured kinematics and those estimated by the forward dynamic simulations using conventional inverse dynamics predicted joint moments and two types of RL agents predicted joint moments

	Conventional inverse dynamics				Kinematics-driven agent				EMG-driven agent			
	RMSE (deg)		Correlation coefficient		RMSE (deg)		Correlation coefficient		RMSE (deg)		Correlation coefficient	
# Subject	Wrist	MCP	Wrist	MCP	Wrist	MCP	Wrist	MCP	Wrist	MCP	Wrist	MCP
1	29.8	14.8	77%	76%	7.9	6.7	99%	97%	13.6	10.6	95%	83%
2	115.5	40.3	57%	45%	12.7	6.9	96%	95%	23.8	15.5	90%	79%
3	59.2	173.8	60%	1%	9.6	12.5	98%	93%	17.9	15.8	95%	87%
4	72.2	58.1	88%	34%	8.9	7.5	99%	95%	16.3	16.1	96%	78%
5	157.4	11.5	45%	85%	14.3	5.4	97%	97%	19.1	11.8	96%	83%
6	80.6	27.6	23%	79%	6.0	11.0	97%	94%	9.9	10.5	91%	95%
Mean (standard deviation)	**85.8 (44.9)**	**54.3 (61.0)**	**58% (23%)**	**53% (33%)**	**9.9 (3.1)**	**8.3 (2.8)**	**98% (1%)**	**95% (2%)**	**16.8 (4.8)**	**13.4 (2.7)**	**94% (03%)**	**84% (6%)**

simulation samples and converging speed [26]. Once trained, the policy of the agent remained unchanged throughout movements.

Validation. We assessed predictions of the trained agents using cross-validation. For the kinematics-driven agent, validation kinematics data were passed to a trained agent to predict corresponding joint moments. Note that during the validation, the trained RL agent was fixed; no additional learning/update of the agent from reward was implemented. Resulting joint moments were smoothed using a local regression filter (i.e., using weighted linear least squares and second-degree polynomial model, spanning 2% of total data points), and drove the kinetic hand model via forward dynamics to predict joint kinematics over time without compensation (i.e., open loop simulation). Predicted kinematics were compared with measured kinematics using Pearson correlation coefficient and root-mean-squared error (RMSE) (Fig. 2). In order to evaluate the performance of the RL trained agent, we also compared it to conventional inverse dynamics for joint moment estimation. Since there was no ground truth of actual joint moments, we also compared the actual kinematics with forward dynamics predicted kinematics over time without compensation using conventional inverse dynamics-estimated joint moment as a driver. The same validation approach was adopted for the EMG-driven agents, except that predicted moments were not smoothed before forward dynamics because they were intended to mimic use in real-time HMI.

Active and Passive Moment-Angle(-Electromyography) Relationship Extraction. We also tested if trained EMG-driven agents could predict other biomechanical features without any physiological knowledge, specifically active moment-angle-EMG and passive moment-angle relationships. The active wrist moment-angle-EMG relationship of each subject was obtained by feeding each trained EMG-driven agent with wrist angles from −80 deg to 80 deg (flexion and extension are positive and negative, respectively) and wrist muscle EMG from −4 to 4 times normalized EMG (negative and positive EMGs represented activation of *extensors and flexors*, respectively), while MCP joint angle and EMGs of the other two muscles were held at zero. Passive wrist and MCP moment–angle relationships were extracted by setting the trained agent's EMG inputs to zero, and the wrist and MCP angles swept from −80 deg to 80 deg and from −5 deg to 70 deg, respectively. The ranges of both joint angles were selected because these were the ranges of motion common among the collected kinematics of the six subjects.

Learning Transfer of Electromyography-Driven Agent. We tested whether knowledge learned by the EMG-driven agent from one subject's data could be transferred to new subjects, thus increasing training speed. Instead of starting with random parameters, we initialized training using a pretrained policy, obtained from the training from another subject at 1×10^6 training steps (i.e., a training step represents an iteration of gradient descent optimization). The relationship between cumulative reward and training step was compared to that of training for the same subject dataset but initialized with random initial parameters.

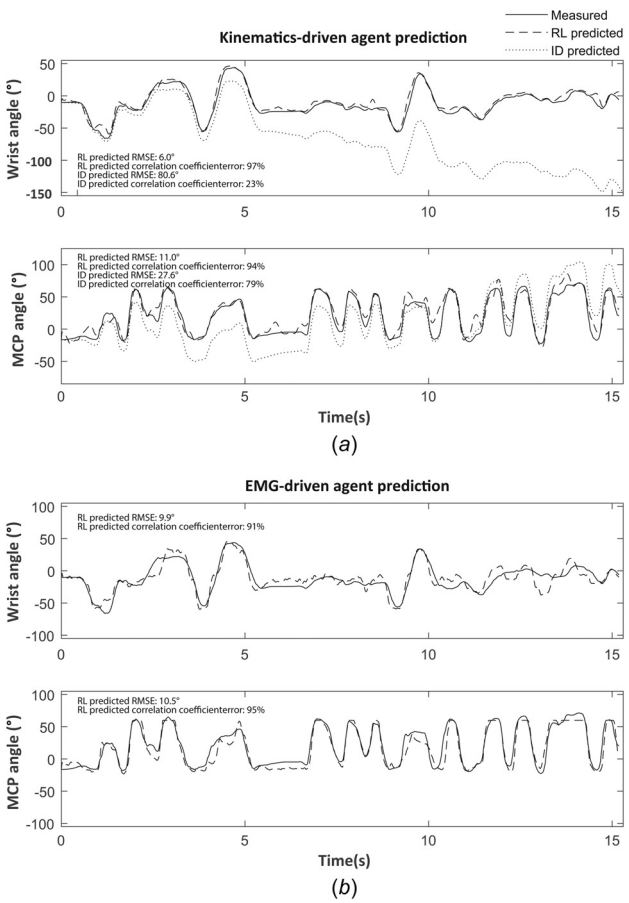

Fig. 3 Comparison of measured kinematics of simultaneous wrist and MCP movement to the those obtained from forward dynamics driven by moments predicted from (*a*) the kinematics-driven RL agent and conventional inverse dynamics (ID), and (*b*) the EMG-driven agent. Example subject dataset is shown. Flexion and extension are noted by positive and negative signs, respectively.

Results

Training time for the kinematics-driven agent was 62 min, with 0.73×10^6 training steps. All EMG-driven agents reached rewards that were greater than 400 within 6 h, equivalent to approximately 3×10^6 training steps.

Figure 2 shows cross-validation workflow for the trained RL agent driven by an example of wrist kinematics. The predicted joint moment remained noisy (Fig. 2(*b*)). However, the smoothing filter removed the noise without compromising much on the kinematics prediction (Figs. 2(*c*) and 2(*d*)). Figure 3(*a*) shows the measured wrist and MCP kinematics and the kinematics predicted by the forward dynamic simulation without closed-loop error compensation, driven by the kinematics-based RL agent and the conventional inverse dynamics-estimated joint moments, respectively. The kinematics predicted by the RL agent was stable within the simulation period and correlated well with measured joint kinematics data, with 98% ± 1% (mean ± standard deviation) for wrist and 95% ± 2% correlation coefficient for MCP (Table 2). The RMSE for wrist and MCP were 9.9 ± 3.1 deg and 8.3 ± 2.8 deg, respectively. In contrast, the kinematics predicted by the forward dynamic simulations without error compensation using conventional inverse dynamics-estimated joint moments tended to drift away from the measured kinematics and became unstable after around 5 s of simulation due to accumulated estimation and computational errors occurring in both inverse and forward dynamics computation (Fig. 3(*a*)). The RMSE and correlation coefficient between them were 85.8 ± 44.9 deg and 0.58 ± 0.23 for the wrist and 54.3 ± 61.0 deg and 0.53 ± 0.33 for the MCP joint, respectively (Table 2).

Figure 3(*b*) shows the measured kinematics and kinematics predicted by forward dynamic simulation without error compensation, driven by the EMG-based RL agent. Across all subject-specific EMG-driven agents, the correlation coefficient between predicted and measured kinematics was 94% ± 3% for the wrist and 84% ± 6% for MCP (Table 2), while RMSE between predicted and measured kinematics were 16.8 deg ± 4.8 deg and 13.4 deg ± 2.7 deg, respectively.

The active and passive moment-angle(-EMG) relationships were extracted from the trained EMG-driven agent. Specifically, the trained EMG-driven agent predicted that wrist joint moments were positively correlated to EMG level and negatively correlated to joint angle (Fig. 4(*a*)). Similarly, predicted passive wrist moment was negatively correlated to joint angle, which was consistent with experimentally reported passive wrist joint moments [29,30] (Fig. 4(*c*)). For passive moment coupling between the wrist and MCP joints, MCP joint angle showed limited influence

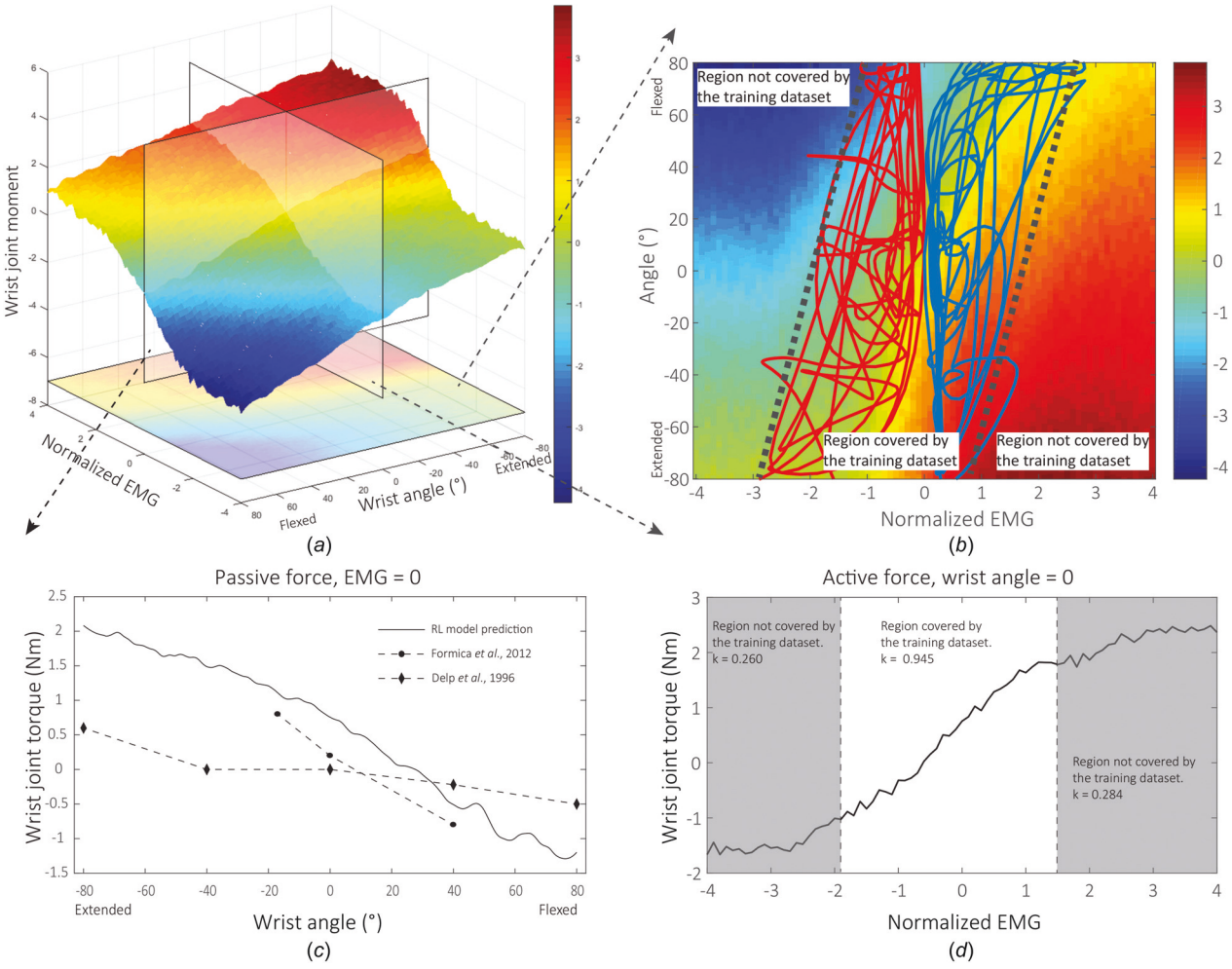

Fig. 4 Joint moment-generating features predicted by example EMG-driven agent. (*a*) RL-predicted joint moment with respect to normalized EMG level and wrist joint angle. (*b*) 2D colormap of the joint moment-EMG-joint angle relationship and dataset used to train the RL agent. The blue and red curves represent the training flexor and extensor muscle EMGs, respectively. (*c*) Predicted passive wrist moment when all actuators are not activated (i.e., EMG = 0). Measurement comparison data points are adapted from Refs. [29] and [30]. and (*d*) Active wrist moment predicted by RL agent when wrist joint angle is 0. The *k* value represents the gradient of the linear regression of the curve in each region.

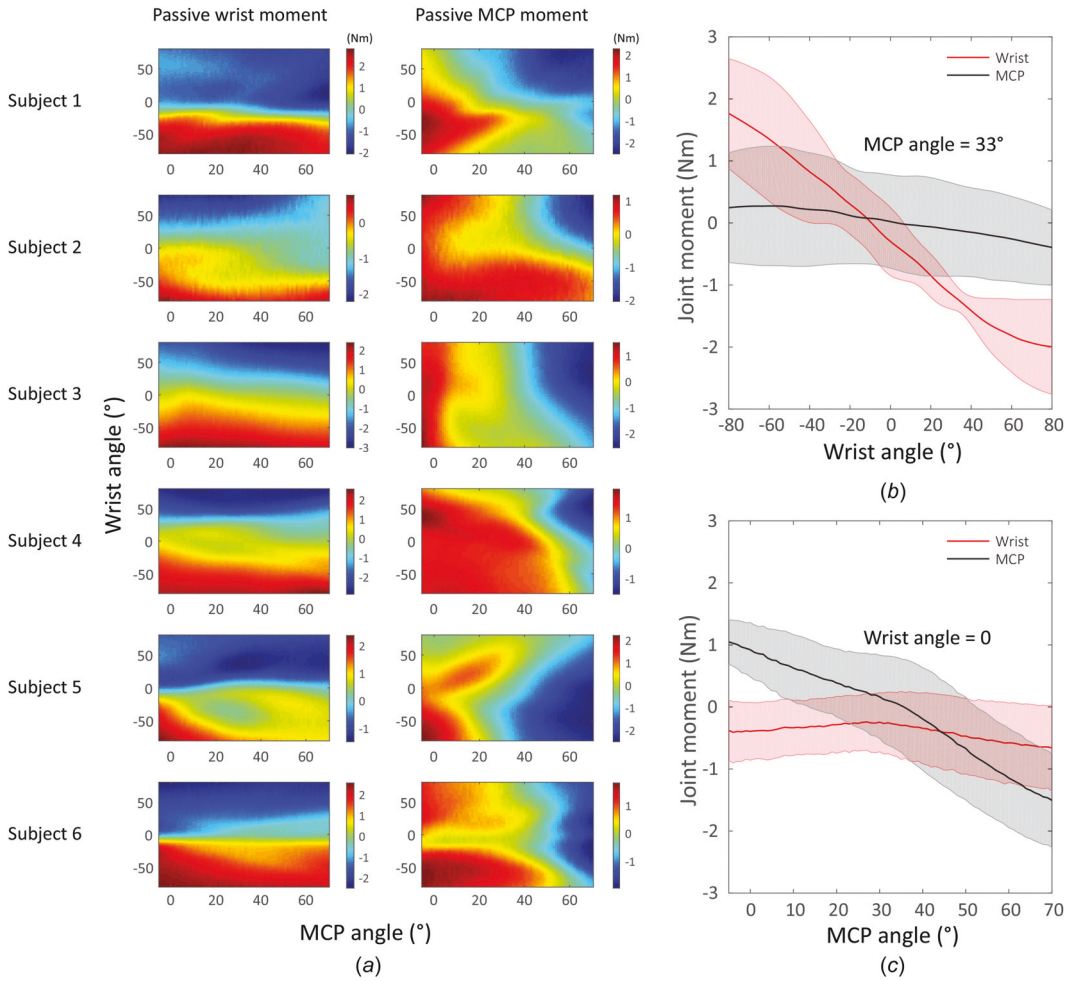

Fig. 5 The passive moment coupling features (i.e., EMGs = 0) predicted by subject-specific EMG-driven agents. (*a*) Colormaps of the passive moments of each subject as the wrist and MCP joint angle varied. (*b*) Passive joint moments when MCP joint was fixed at the center point of the movement range (i.e., 33 deg) as wrist joint varied. (*c*) Passive joint moment when wrist joint was fixed at the center point of the movement range (i.e., 0 deg) as MCP joint varied. The solid line and the shaded area represent the mean and standard deviation over the six agents.

on predicted passive wrist moment for all subjects, while both wrist joint and MCP joint angles were generally negatively correlated to the passive MCP joint moment (Fig. 5).

The collected training data only covered a portion of the possible posture-EMG space (Fig. 4(*b*)), due to the nature of a free hand movement. Active moment features predicted by trained RL agents depended on the training data coverage. For example, when wrist angle was zero, the linear regression gradient of the moment-EMG curve was 0.945 in the region covered by the training dataset (i.e., unshaded region in Fig. 4(*d*)) while they were 0.260 and 0.284 outside the training dataset (i.e., shaded region in Fig. 4(*d*)).

The knowledge learned by the RL agent from one subject's dataset improved training speed when transferred to a new subject dataset. When the EMG-driven agent was initialized with a predefined policy that was trained on the dataset from another subject, it still started with low cumulative reward, but the learning speed

Table 3 Summary of advantages and disadvantages of the RL agent

	Description
Advantage	Avoided the error introduced by the second-order motion differentiation
	Had relatively fast optimization speed compared to other forward-dynamics-based optimizations
	Enabled learning transfer between subjects during training
	Had better EMG error tolerance and made EMG excitation adjustment un-necessary
	Enabled easy sensor expandability
	Extracted informative subject-specific joint moment generating feature
	Provided an additional layer of information for data-driven HMI tools
Disadvantages	Relied on the scope of the training dataset
	Still needed inertia properties for inverse dynamics
	Had longer optimization time compared to optimizations that do not require forward dynamics, such as pattern recognition and MSK parameters optimization by matching measured the joint moment

was 3.6 times faster than when initialized with random parameters (Fig. 6).

Discussion

In this study, we proposed an RL method, as an alternative technique to conventional MSK-based approaches, to predict joint moments based on either measured kinematics or surface EMGs during free MCP and wrist movement. Estimated joint moments from each RL agent used forward dynamics simulation (without closed-loop compensation) to predict kinematics. Impressively within the simulation period (15 s), both RL agents can closely approximate measured kinematics via open-loop simulation in cross-validation. This suggests the proposed RL method is feasible to either provide an alternative approach to inverse dynamics analysis or potentially be applied as an HMI tool. Reasonable subject-specific joint moment-generating features can also be estimated from the trained RL agents without physiological knowledge, but it depends on the range of trustable scope and variations covered by the training data. Advantages and disadvantages of our RL approach (Table 3) are discussed below.

The kinematics-driven agent predicted the joint moment, and subsequently joint kinematics via open-loop forward dynamics simulation, more accurately and stably than the conventional inverse dynamics (Fig. 3(*a*) and Table 2). This is because the conventional ID method tends to amplify the kinematics measurement noise during second-order differentiation when external force measurements are unavailable, and the computation errors in both inverse dynamics and forward dynamic processes accumulate over time if no compensation is applied [5]. The open-loop simulation became unstable after around 5 s of simulation. The RL-based kinematics-driven agent, on the other hand, showed more robust and stable performance against these errors in the cross validation within the simulation period due to the formulation of reward function during policy learning. Yet, if a specific biomechanical application requires high accuracy for tracing given kinematics during forward dynamic simulation, additional feedback control is needed to compensate the motion predication errors observed in open-loop simulation. For the EMG-driven agent, our approach yields comparable or better performance (i.e., higher Pearson correlation coefficient and lower RMSE), compared to the existing EMG-based HMI for continuous estimation of joint motion during offline analysis, such as linear regression, artificial neural network, and lumped parameter musculoskeletal model [31]. It has been suggested that the correlation coefficient between measured and predicted kinematics is more accurate indicator for closed-loop HMI performance than the RMSE [32]. Specifically, a closed-loop HMI operation requires accurate prediction the user's intended movement direction and speed for intuitive device control, which can be reflected by the correlation coefficient. In contrast, the RMSE may be introduced by the errors accumulated over time during the forward dynamics process offline. It can be greatly mitigated in a closed-loop, real-time operation with a human in the loop. This is because human operator (controller) can fine tune muscle contraction based on visual/haptic feedback to compensate error between the intended and actual position. Thus, one of our future works is to implement and evaluate our developed EMG-driven agent in closed-loop HMI applications.

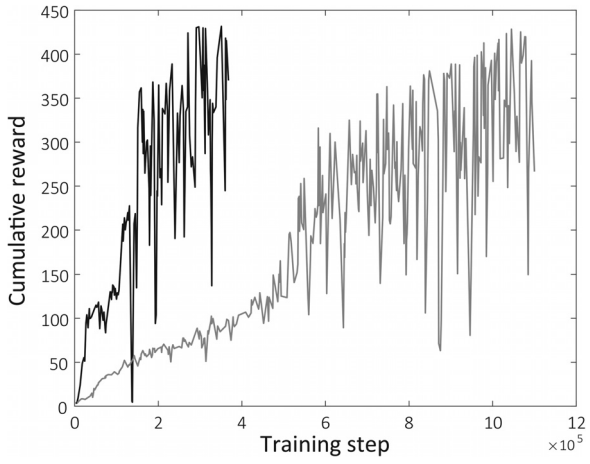

Fig. 6 Cumulative reward of EMG-driven agent with respect to training step. When training was initialized with random parameters (gray), the final reward was reached more slowly than when the agent was trained with the same dataset but initialized with a policy pretrained from another subject's dataset (black).

The RL method offers an innovative approach to obtain joint moment from kinematics without using external force measurements. In contrast to the standard Newton's-Law-based inverse dynamics that uses instantaneous timesteps and exact kinematics measurements, the presented RL approach predicts joint moment by using future timesteps and a range of kinematics within an error threshold. Similar to the scenario of machine learning-based autonomous driving [33], this allows the RL agent to learn to "drive" the hand kinetic model along the "road" of measured kinematics, where future kinematics are like the road ahead and the error threshold is like road width. Though the current application is a simple planar hand model, this method has potential to extend to a more complex system in the future, such as gait analysis with limited or alternative (e.g., insole pressure sensor) ground reaction force measurements.

The EMG-driven agent not only showed good prediction of kinematics during cross-validation but also exhibited relatively fast training speed compared to typical parameter optimization. Training the EMG-driven agent to reach the saturated reward without any previous knowledge took <6 h for all participants, while optimization time even for a lumped-actuator MSK model was >10 h in our previous study using a similar processor [2]. Fast training speed is achieved because: (1) in each iteration, forward dynamics through the whole training data time range is not required before the policy is updated. A nonoptimal policy is likely to drive the kinetic model outside of the error threshold and thus reset the episode early in the time range (Fig. 1). In contrast, optimization approaches to obtain subject-specific MSK musculotendon parameters via matching measured kinematics require forward dynamics analysis of the whole dataset in each iteration. (2) The training agent can learn from an array of hand kinetic models that run forward dynamics in parallel, increasing the number of learning sources and speeding up training. Additionally, we demonstrated that RL training time can be further reduced if training is initiated with a pretrained policy. Even if the policy is trained from datasets of other subjects, it contains basic system information (e.g., general EMG-force relationship), reducing iteration number.

Other benefits of an EMG-driven agent include better EMG error tolerance and easy sensor expandability. First, surface EMG collected from the forearm is likely to be affected by EMG crosstalk due to small muscle size [9]. Though EMG crosstalk is usually considered as noise, if the crosstalk is consistent, it can be beneficial to data-driven RL because it amplifies signal magnitude. EMG adjustment adopted by Hoang et al. (2018) is also unnecessary because it is embedded in the trained RL policy [10]. Second, new measured inputs can be easily added. When extra inputs (e.g., additional EMG sensors, accelerometers) are added to data collection, it is easy to include their data in the RL system for performance improvement, without manually interpreting physical meaning of the data or altering the kinetic model.

The proposed RL method uncovers meaningful information describing subject-specific joint moment features using only a small amount (i.e., 15 s) of measured data without any knowledge of underlying physiological structure. For example, the simulations demonstrated that predicted passive wrist moment was greatly influenced by wrist posture but not MCP joint posture (Figs. 4(*c*) and 5), while both wrist joint and MCP joint angles

influenced passive MCP joint moment. This reflects observed behavior of the physiological system [34]. Here, we present example characteristics that can be elucidated using this RL approach; additional features such as moment–joint angular velocity relationship, and active wrist and MCP coupling can also be easily obtained by feeding the trained RL agent with relevant joint and muscle states. This approach can identify important functional behaviors for specific subjects, which has potential to assist in MSK model development, medical diagnosis, and rehabilitation progress assessment.

The proposed RL method provides researchers with additional information for data-driven HMI tools. Though data-driven approaches have shown great strength in designing HMI tools [35,36], many function as black-boxes. For example, pattern recognition—one of the most studied approaches in EMG-driven devices—maps measured EMG signals to prescribed motions with high classification accuracy [37], but the physical relationship between muscle activation, joint moment, and joint motion is ignored by the mapping. This results in difficulties configuring the system and compromises system robustness—minor noise in an EMG signal may result in unexpected motion results [36]. Using the trained agent to predict joint moment and subsequently derive joint kinematics using forward dynamics, however, can mitigate such issues because (1) the agent predicts reasonably smooth moment features (Fig. 4), where minor noise is unlikely to result in sudden unexpected changes in behavior; (2) if there are unexpected kinematic outcomes, it is more intuitive for researchers to identify the problem by examining predicted moment features.

One major limitation of the EMG-driven agent is that performance relies on the training dataset scope. The RL method is unable to predict joint moment well when input EMG and joint angle are outside the training dataset range. In contrast, an MSK model has inherent knowledge of underlying MSK structure and EMG–force relationship, so is able to predict reasonable results even with a generic model [32]. However, the predefined and simplified structure may potentially limit such a model from capturing moment-generating behavior of a specific subject. We therefore suggest that future studies could potentially combine an RL agent and MSK model to design a mutually complementary EMG-driven controller—for example, using the RL method when input states are within the training region to provide finer control, and using the MSK model when input states are beyond the training region to provide physiologically based force estimations. Indeed, including more variations in the training data and increasing training time can potentially improve performance of the agents, and understanding the tradeoffs between them is critical future work. It would also be valuable in the future to study the robustness of EMG-driven RL agent against EMG variations caused by physical or physiological changes over time (e.g., muscle fatigue) in order to apply it for HMI applications. Interestingly, previous studies showed robustness of EMG-based HMIs against the EMG variations for continuous motion estimation with real-time, human-in-the-loop testing [32,38–40]. This is partly because of human adaptation; end users can instantly modify the level of effort to compensate for the variations of EMG interface. Therefore, in our future study, we postulate that our RL engine, when used as EMG-based HMI with human-in-the-loop, is robust against a certain level of EMG signal variations. Our study is also limited by the number of human subjects ($n = 6$) tested because this technical brief only served as a proof of concept of using reinforcement learning to predict joint moment. Despite the variations across the participants, we demonstrated the proposed technique was able to capture salient features of each subject by customizing the RL policies based on each individual person's data. In the future, when our approach is used to evaluate human biomechanical characteristics, more human subjects are needed. Furthermore, we only tested a single learning transfer case between a pair of EMG-driven agents, yet we believe a more thorough investigation that requires repeated testing over multiple subjects and conditions (, e.g., initial conditions with different pretrained steps and different ending conditions) is needed to better explore the potentials. One limitation of the kinematics-driven agent is that although error from motion differentiation is avoided, accuracy of moment estimates still relies on estimated inertia; a generic model was used here without subject-specific inertias. Both agents were tested in a low-inertia regime with simple two degree-of-freedom planar motion; future systematic examination of RL approach on more complex systems is needed.

In conclusion, this study illustrates that an RL approach can be an alternative technique to conventional inverse dynamic analysis in human biomechanics study and EMG-driven HMI applications. The study also illustrated that RL can reveal specific subject's joint moment-generating features. Future work will extend to more complex systems like gait analysis and systematically examine integration RL method with MSK models.

Funding Data

- NSF (Grant Nos. #1527202, 1637892, and 1856441; Funder ID: 10.13039/100000001).
- DOD (Grant Nos. #W81XWH-15-C- 0125 and W81XWH-15-1-0407; Funder ID: 10.13039/100000005).

References

[1] Zajac, F. E., 1989, "Muscle and Tendon: Properties, Models, Scaling, and Application to Biomechanics and Motor Control," Crit. Rev. Biomed. Eng., **17**(4), pp. 359–411.

[2] Crouch, D. L., and Huang, H., 2016, "Lumped-Parameter Electromyogram-Driven Musculoskeletal Hand Model: A Potential Platform for Real-Time Prosthesis Control," J. Biomech., **49**(16), pp. 3901–3907.

[3] Holzbaur, K. R. S., Murray, W. M., and Delp, S. L., 2005, "A Model of the Upper Extremity for Simulating Musculoskeletal Surgery and Analyzing Neuromuscular Control," Ann. Biomed. Eng., **33**(6), pp. 829–840.

[4] Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., and Thelen, D. G., 2007, "OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement," IEEE Trans. Biomed. Eng., **54**(11), pp. 1940–1950.

[5] Kuo, A. D., 1998, "A Least-Squares Estimation Approach to Improving the Precision of Inverse Dynamics Computations," ASME J. Biomech. Eng., **120**(1), pp. 148–159.

[6] Otten, E., 2003, "Inverse and Forward Dynamics: Models of Multi–Body Systems," Philos. Trans. R. Soc. London. Ser. B Biol. Sci., **358**(1437), pp. 1493–1500.

[7] Sartori, M., Llyod, D. G., and Farina, D., 2016, "Neural Data-Driven Musculoskeletal Modeling for Personalized Neurorehabilitation Technologies," IEEE Trans. Biomed. Eng., **63**(5), pp. 879–893.

[8] Farina, D., and Negro, F., 2012, "Accessing the Neural Drive to Muscle and Translation to Neurorehabilitation Technologies," IEEE Rev. Biomed. Eng., **5**, pp. 3–14.

[9] Boostani, R., and Moradi, M. H., 2003, "Evaluation of the Forearm EMG Signal Features for the Control of a Prosthetic Hand," Physiolog. Meas., **24**(2), pp. 309–319.

[10] Hoang, H. X., Pizzolato, C., Diamond, L. E., and Lloyd, D. G., 2018, "Subject-Specific Calibration of Neuromuscular Parameters Enables Neuromusculoskeletal Models to Estimate Physiologically Plausible Hip Joint Contact Forces in Healthy Adults," J. Biomech., **80**, pp. 111–120.

[11] Lloyd, D. G., and Besier, T. F., 2003, "An EMG-Driven Musculoskeletal Model to Estimate Muscle Forces and Knee Joint Moments In Vivo," J. Biomech., **36**(6), pp. 765–776.

[12] Garner, B. A., and Pandy, M. G., 2003, "Estimation of Musculotendon Properties in the Human Upper Limb," Ann. Biomed. Eng., **31**(2), pp. 207–220.

[13] Wu, W., Lee, P. V. S., Bryant, A. L., Galea, M., and Ackland, D. C., 2016, "Subject-Specific Musculoskeletal Modeling in the Evaluation of Shoulder Muscle and Joint Function," J. Biomech., **49**(15), pp. 3626–3634.

[14] Kober, J., Bagnell, J. A., and Peters, J., 2013, "Reinforcement Learning in Robotics: A Survey," Int. J. Rob. Res., **32**(11), pp. 1238–1274.

[15] Nguyen, H., and La, H., 2019, "Review of Deep Reinforcement Learning for Robot Manipulation," Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, Feb. 25–27, pp. 590–595.

[16] Kormushev, P., Calinon, S., and Caldwell, D. G., 2013, "Reinforcement Learning in Robotics: Applications and Real-World Challenges," Robotics, **2**(3), pp. 122–148.

[17] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D., 2016, "Mastering the Game of Go With Deep Neural Networks and Tree Search," Nature, **529**(7587), pp. 484–489.

[18] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D., 2015, "Human-Level Control Through Deep Reinforcement Learning," Nature, **518**(7540), pp. 529–533.
[19] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Van Den Driessche, G., Graepel, T., and Hassabis, D., 2017, "Mastering the Game of Go Without Human Knowledge," Nature, **550**(7676), pp. 354–359.
[20] Kidziński, Ł., Mohanty, S. P., Ong, C., Huang, Z., Zhou, S., Pechenko, A., Stelmaszczyk, A., Jarosik, P., Pavlov, M., Kolesnikov, S., Plis, S., Chen, Z., Zhang, Z., Chen, J., Shi, J., Zheng, Z., Yuan, C., Lin, Z., Michalewski, H., Miłoś, P., Osiński, B., Melnik, A., Schilling, M., Ritter, H., Carroll, S., Hicks, J., Levine, S., Salathé, M., and Delp, S., 2018, "Learning to Run Challenge Solutions: Adapting Reinforcement Learning Methods for Neuromusculoskeletal Environments," The NIPS'17 Competition: Building Intelligent Systems, Springer, Berlin, pp. 121–153.
[21] Kidziński, Ł., Mohanty, S. P., Ong, C., Hicks, J. L., Carroll, S. F., Levine, S., Salathé, M., and Delp, S. L., 2018, "Learning to Run Challenge: Synthesizing Physiologically Accurate Motion Using Deep Reinforcement Learning," The NIPS'17 Competition: Building Intelligent Systems, Springer, Berlin, pp. 101–120.
[22] Pavlov, M., Kolesnikov, S., and Plis, S. M., 2017, "Run, Skeleton, Run: Skeletal Model in a Physics-Based Simulation," arXiv preprint arXiv:1711.06922.
[23] Wen, Y., Si, J., Brandt, A., Gao, X., and Huang, H., 2020, "Online Reinforcement Learning Control for the Personalization of a Robotic Knee Prosthesis," IEEE Trans. Cybern., **50**(6), pp. 2346–2356.
[24] Wen, Y., Si, J., Gao, X., Huang, S., and Huang, H. H., 2017, "A New Powered Lower Limb Prosthesis Control Framework Based on Adaptive Dynamic Programming," IEEE Trans. neural Networks Learn. Syst., **28**(9), pp. 2215–2220.
[25] Jagodnik, K. M., Thomas, P. S., van den Bogert, A. J., Branicky, M. S., and Kirsch, R. F., 2017, "Training an Actor-Critic Reinforcement Learning Controller for Arm Movement Using Human-Generated Rewards," IEEE Trans. Neural Syst. Rehabil. Eng., **25**(10), pp. 1892–1905.
[26] Juliani, A., Berges, V.-P., Vckay, E., Gao, Y., Henry, H., Mattar, M., and Lange, D., 2018, "Unity: A General Platform for Intelligent Agents," arXiv preprint arXiv:1809.02627.
[27] Ramachandran, P., Zoph, B., and Le, Q. V., 2017, "Swish: A Self-Gated Activation Function," 6th International Conference on Learning Representations, Vancouver, BC, Canada, Apr. 30–May 3.
[28] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., 2017, "Proximal Policy Optimization Algorithms," arXiv preprint arXiv1707.06347.
[29] Formica, D., Charles, S. K., Zollo, L., Guglielmelli, E., Hogan, N., and Krebs, H. I., 2012, "The Passive Stiffness of the Wrist and Forearm," J. Neurophysiol., **108**(4), pp. 1158–1166.
[30] Delp, S. L., Grierson, A. E., and Buchanan, T. S., 1996, "Maximumisometric Moments Generated by the Wrist Muscles in Flexion-Extension and Radial-Ulnar Deviation," J. Biomech., **29**(10), pp. 1371–1375.
[31] Pan, L., Crouch, D. L., and Huang, H., 2019, "Comparing EMG-Based Human-Machine Interfaces for Estimating Continuous, Coordinated Movements," IEEE Trans. Neural Syst. Rehabil. Eng., **27**(10), pp. 2145–2154.
[32] Pan, L., Crouch, D. L., and Huang, H., 2018, "Myoelectric Control Based on a Generic Musculoskeletal Model: Toward a Multi-User Neural-Machine Interface," IEEE Trans. Neural Syst. Rehabil. Eng., **26**(7), pp. 1435–1442.
[33] Chen, C., Seff, A., Kornhauser, A., and Xiao, J., 2015, "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving," Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, Dec. 7–13, pp. 2722–2730.
[34] Knutson, J. S., Kilgore, K. L., Mansour, J. M., and Crago, P. E., 2000, "Intrinsic and Extrinsic Contributions to the Passive Moment at the Metacarpophalangeal Joint," J. Biomech., **33**(12), pp. 1675–1681.
[35] Quitadamo, L. R., Cavrini, F., Sbernini, L., Riillo, F., Bianchi, L., Seri, S., and Saggio, G., 2017, "Support Vector Machines to Detect Physiological Patterns for EEG and EMG-Based Human–Computer Interaction: A Review," J. Neural Eng., **14**(1), p. 011001.
[36] Iqbal, N. V., Subramaniam, K., and P, S. A., 2018, "A Review on Upper-Limb Myoelectric Prosthetic Control," IETE J. Res, **64**(6), pp. 740–752.
[37] Scheme, E., and Englehart, K., 2011, "Electromyogram Pattern Recognition for Control of Powered Upper-Limb Prostheses: State of the Art and Challenges for Clinical Use," J. Rehabil. Res. Dev., **48**(6), pp. 643–660.
[38] Pradhan, A., Jiang, N., Chester, V., and Kuruganti, U., 2020, "Linear Regression With Frequency Division Technique for Robust Simultaneous and Proportional Myoelectric Control During Medium and High Contraction-Level Variation," Biomed. Signal Process. Control, **61**, p. 101984.
[39] Pan, L., Harmody, A., and Huang, H., 2018, "A Reliable Multi-User EMG Interface Based on a Generic-Musculoskeletal Model Against Loading Weight Changes," 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, July 18–21, pp. 2104–2107.
[40] Hahne, J. M., Schweisfurth, M. A., Koppe, M., and Farina, D., 2018, "Simultaneous Control of Multiple Functions of Bionic Hand Prostheses: Performance and Robustness in End Users," Sci. Robot., **3**(19), p. eaat3630.